SuiteCloud CLI for Java
should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and "as available", for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer's use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle's other customers, Oracle may slow down or throttle Customer's excess use until such time that Customer's use stays within reasonable limits. If Customer's particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer's actual usage pattern.

Beta Features

Oracle may make available to Customer certain features that are labeled “beta” that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party's property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle's standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle's customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a “beta” label.
Send Us Your Feedback

We'd like to hear your feedback on this document.

Answering the following questions will help us improve our help content:

- Did you find the information you needed? If not, what was missing?
- Did you find any errors?
- Is the information clear?
- Are the examples correct?
- Do you need more examples?
- What did you like most about this document?

Click [here](#) to send us your comments. If possible, please provide a page number or section title to identify the content you’re describing.

To report software issues, contact NetSuite Customer Support.
# Table of Contents

CLI for Java Guide .................................................................................................................. 1  
CLI for Java Installation Prerequisites ..................................................................................... 1  
Installing CLI for Java ........................................................................................................... 1  
Token-based Authentication in CLI for Java ............................................................................. 2  
SDF File Structure ................................................................................................................. 3  

CLI for Java Reference ............................................................................................................. 5  
CLI for Java Commands ......................................................................................................... 5  
  adddependencies ........................................................................................................... 6  
  authenticate .................................................................................................................. 9  
  createproject ............................................................................................................... 10  
  deploy ........................................................................................................................ 11  
  importbundle ............................................................................................................... 14  
  importconfiguration ...................................................................................................... 15  
  importfiles ................................................................................................................... 17  
  importobjects .............................................................................................................. 19  
  issuetoken ................................................................................................................... 21  
  listbundles ................................................................................................................... 23  
  listconfiguration ........................................................................................................... 24  
  listfiles ........................................................................................................................ 26  
  listmissingdependencies ............................................................................................... 27  
  listobjects .................................................................................................................... 29  
  manageauth ................................................................................................................ 31  
  preview ........................................................................................................................ 32  
  project ........................................................................................................................ 34  
  revoketoken ................................................................................................................. 35  
  savetoken ..................................................................................................................... 36  
  update ........................................................................................................................ 38  
  updatecustomrecordwithinstances .................................................................................. 40  
  uploadfiles .................................................................................................................... 41  
  uploadfolders ............................................................................................................... 43  
  validate ....................................................................................................................... 44
CLI for Java Guide

SuiteCloud Command Line Interface for Java (SuiteCloud CLI for Java) is a tool that can be used with your own integrated development environment (IDE) to develop SDF projects. You can also create batch and shell scripts that use CLI commands to automate SDF project validation and deployment processes. CLI for Java is part of the SuiteCloud Software Development Kit (SuiteCloud SDK).

Note: CLI for Java is intended for experienced SDF users and users who are already familiar with SuiteCloud IDE plug-ins. Oracle recommends using SuiteCloud IDE plug-ins to develop your first project and to learn more about SDF before trying CLI for Java. For more information, see the help topics Getting Started with SuiteCloud Development Framework and SuiteCloud Development Framework Tutorial.

The following topics provide instructions to install and use CLI for Java:

- CLI for Java Installation Prerequisites
- Installing CLI for Java
- Token-based Authentication in CLI for Java
- .SDF File Structure

CLI for Java Installation Prerequisites

Verify the following prerequisites before you install use CLI for Java:

- **File extractor software** — File extractor software is required to extract the contents of the installation package.
- **Java Runtime Environment (JRE)** — CLI for Java requires Java SE Development Kit version 11.
- **NetSuite account features** — An account administrator must enable the features in your account that SDF depends on. See the help topic Enabling SDF in Your NetSuite Account (Admin Only).
- **NetSuite permissions** — An account administrator must create and assign the SDF Developer role to your account. See the help topic Assigning the Developer Role (Admin Only).
- **Operating system** — CLI for Java supports the following operating systems:
  - Windows
  - Linux
  - Mac OS

Installing CLI for Java

To install CLI for Java, extract the installation package so that you can then execute CLI using Java.

Important: When your account is upgraded, your instance of CLI for Java does not get updated automatically. After your Account upgrade, you must manually update CLI for Java to match the version of your Account. You may experience errors related to compatibility by using an older version of CLI for Java with the latest version of NetSuite.

To install CLI for Java:

1. Create a directory where you want to install CLI for Java, such as C:\javacli\ for Windows, or /javacli/ for Linux and Mac OS X.
2. Adjust your PATH system environment variable to include the CLI for Java directory. By doing this, you are enabling the use of CLI for Java from any folder in your machine.
   - Select Control Panel > System > Advanced system settings.
   - Click Environment Variables...
   - Under System variable, select the PATH variable, and click Edit...
   - Click New, enter the CLI for Java directory path, and click OK.
   - In the Environment Variables window, click OK. Then, in the System Properties window, click Apply, and then OK.

3. Download CLI for Java and extract the contents of one of the following packages to the CLI for Java directory:
   - If your account runs on 2020.1, use one of the following links:
     - Linux and Mac: https://system.netsuite.com/download/suitecloud-sdk/cli/java/latest/sdf-cli.tar.gz
     - Note: Use this link to get the latest changes every time the CLI for Java is updated.
   - If your account runs on 2019.2, use one of the following links:
     - Linux and Mac: https://system.netsuite.com/download/suitecloud-sdk/cli/java/19.2/sdf-cli.tar.gz
     - Note: The packages contain the JAR file and either the CLI for Java bash or batch script, depending on the operating system.

   You can now use CLI for Java by running the sdfcli command.

4. For Linux and Mac OS X computers, make the sdfcli bash file executable by running the following command from the CLI for Java directory:

   ```bash
   chmod +x sdfcli
   ```

   You can now use CLI for Java by running the sdfcli command.

   - Note: You must have read, write, and execute access on any directory where you want to run CLI for Java.

---

## Token-based Authentication in CLI for Java

Token-based authentication (TBA) can be used with custom roles to run CLI for Java commands without being prompted for account passwords. By using TBA with CLI for Java, you can create batch and shell scripts to automate SDF project validation and deployment processes. Additionally, TBA provides increased development environment security, and fewer session terminations when using a single login to develop SDF projects and navigate NetSuite. For more information about setting up TBA with your account, see the help topic Getting Started with Token-based Authentication.

When your account and role are set up for TBA, you can either issue a token from NetSuite or CLI for Java. Depending on your preferences, see one of the following sections:

- Tokens Issued in NetSuite
Tokens Issued in NetSuite

If you already issued a token in NetSuite and want to use it to log in to your account, you can use the `savetoken` command. It retrieves the latest token created from your account. To revoke it, you can either use the `revoketoken` command or go to NetSuite. For more information, see:

- `savetoken`
- `revoketoken`
- Manage TBA Tokens in the NetSuite UI

Tokens Issued in CLI for Java

If you have not issued a token for your account in NetSuite yet, you can issue it directly in CLI for Java. The `issuetoken` command automatically creates a TBA token in your account for use with CLI for Java. In case you need to revoke a token, you can use the `revoketoken` command, which removes the latest TBA token that was created for your account. You can use `issuetoken` to create multiple TBA tokens, but you can only use `revoketoken` to remove the latest issued TBA token. To remove any undesired TBA tokens, go to NetSuite. For more information, see:

- `issuetoken`
- `revoketoken`
- Manage TBA Tokens in the NetSuite UI

Important: TBA tokens can be assigned to multiple accounts, users, and roles, but you can only have one TBA token per account-user-role combination. For example, assume that you issued three TBA tokens for an account named WolfeElectronics with a role named SDF Developer that is assigned to a user named awolfe@wolfeelectronics.com. In this scenario, the awolfe@wolfeelectronics.com user can only use the latest-issued TBA token with CLI for Java to log in to the WolfeElectronics account. The first two TBA tokens should be manually removed from the account using the NetSuite interface if they were not formally revoked before issuing the latest token.

.SDF File Structure

The SDF file stores account credentials for your project. It must be located in the root directory of your project. This file enables CLI for Java to run certain commands without having to enter account credentials each time. To determine whether a command uses the properties that you can specify in SDF files, see the documentation for that command. For the SDF file to work properly, you need to specify all the properties listed below.

The SDF file stores the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>The ID of your account.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>email</td>
<td>The e-mail address to use to log in to your account.</td>
</tr>
<tr>
<td>role</td>
<td>The internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI for Java. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>url</td>
<td>The URL used to access your account.</td>
</tr>
</tbody>
</table>

**Note:** CLI for Java prompts you to enter account passwords. Passwords cannot be specified in the SDF file.

### Example

The following is an example of SDF file contents:

```
account=000071
email=nlbuild@netsuite.com
role=1014
url=system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the **Show Internal IDs** Preference.
CLI for Java Reference

CLI for Java Commands

You can use CLI for Java to manage SDF project components and validate and deploy projects to your NetSuite account.

**Note:** CLI for Java is intended for experienced SDF users and users who are already familiar with SuiteCloud IDE plug-in. It is recommended that you use SuiteCloud IDE plug-in to develop your first project and to learn more about SDF before trying CLI for Java. For more information, see the help topics Getting Started with SuiteCloud Development Framework and SuiteCloud Development Framework Tutorial.

You can use the SDF file to specify account credentials instead of entering them each time you run commands. For more information, see .SDF File Structure. However, CLI for Java options (such as -account, -email, -role, and -url) override credentials specified in the SDF file. CLI for Java always prompts you to enter your account password unless you log in with token-based authentication.

**Syntax**

```
 sdfcli <command> <option> <argument>
```

**Important:** If an argument is enclosed in angle brackets in the CLI, i.e. `<argument>`, it means that it is mandatory. Otherwise, if an argument is enclosed in brackets, i.e. `[argument]`, it means that it is optional.

**Command Reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adddependencies</td>
<td>Adds missing dependencies to the manifest file.</td>
</tr>
<tr>
<td>authenticate</td>
<td>Grants authorization for SuiteCloud SDK to access NetSuite with an account-role combination.</td>
</tr>
<tr>
<td>createproject</td>
<td>Creates an SDF project.</td>
</tr>
<tr>
<td>deploy</td>
<td>Deploys the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>importbundle</td>
<td>Imports a customization bundle from your NetSuite account and converts it to an account customization project.</td>
</tr>
<tr>
<td>importconfiguration</td>
<td>Imports the feature configuration from your NetSuite account to the account customization project so that you can enable or disable features in the project.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>importfiles</td>
<td>Imports files from your NetSuite account to the account customization project.</td>
</tr>
<tr>
<td>importobjects</td>
<td>Imports custom objects from your NetSuite account to the SDF project.</td>
</tr>
<tr>
<td>issuetoken</td>
<td>Issues a TBA token to authenticate against your NetSuite Production account.</td>
</tr>
<tr>
<td>listbundles</td>
<td>Lists the customization bundles that were created in your NetSuite account.</td>
</tr>
<tr>
<td>listconfiguration</td>
<td>Lists the configuration types that are available in your NetSuite account.</td>
</tr>
<tr>
<td>listfiles</td>
<td>Lists the files in the File Cabinet of your NetSuite account.</td>
</tr>
<tr>
<td>listmissingdependencies</td>
<td>Lists the missing dependencies in the SDF project.</td>
</tr>
<tr>
<td>listobjects</td>
<td>Lists the custom objects in your NetSuite account.</td>
</tr>
<tr>
<td>manageauth</td>
<td>Manages authentication IDs (authid) for all your projects.</td>
</tr>
<tr>
<td>preview</td>
<td>Previews the deployment steps of a folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>project</td>
<td>Sets the default project folder or zip file for CLI for Java.</td>
</tr>
<tr>
<td>revoketoken</td>
<td>Revokes the latest TBA token that was issued to your NetSuite Production account using the issuetoken command.</td>
</tr>
<tr>
<td>savetoken</td>
<td>Saves the latest TBA token created from your account to use with CLI for Java.</td>
</tr>
<tr>
<td>update</td>
<td>Updates existing custom objects in the SDF project folder with the custom objects in your NetSuite account.</td>
</tr>
<tr>
<td>updatecustomrecordwithinstances</td>
<td>Updates the custom record object and its instances in the SDF project.</td>
</tr>
<tr>
<td>uploadfiles</td>
<td>Uploads one or several files from an SDF project to the file cabinet in your account.</td>
</tr>
<tr>
<td>uploadfolders</td>
<td>Uploads all the files within one or several folders from an SDF project to the file cabinet in your account.</td>
</tr>
<tr>
<td>validate</td>
<td>Validates the folder or zip file that contains the SDF project.</td>
</tr>
</tbody>
</table>

**adddependencies**

Adds missing dependencies to the manifest file.

**Syntax**

```
adddependencies [-account AccountID] 
[-authid myNetSuiteDevRole] 
[-email EmailAddress] 
[-p FolderOrZipFile] 
[-role InternalID] 
[-url NetSuiteDomainURL] 
[-all] 
[-feature FeatureName1:(optional|required) [FeatureName2:(optional|required)] [...] 
[-file File1 [File2] [...]]
```
CLI for Java Commands

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the email address to use to log in to the account. When specified, the email address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the URL used to access the NetSuite account. When specified, the URL value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-all</td>
<td>Optional</td>
<td>Adds all missing dependencies to the SDF project. When specified, ensure that you do not use the -feature, -file, and -object options.</td>
</tr>
<tr>
<td>-feature</td>
<td>Optional</td>
<td>Specifies the feature dependency references to add to the SDF project. Each reference is a name:value pair delimited by a space. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CUSTOMRECORD:required ACCOUNTING:optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When specified, ensure that you do not use the -all option.</td>
</tr>
</tbody>
</table>
### CLI for Java Commands

<table>
<thead>
<tr>
<th>Argument</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-file</code></td>
<td></td>
<td>Specifies the files to add to the SDF project. When specified, ensure that you do not use the <code>-all</code> option.</td>
</tr>
<tr>
<td><code>-object</code></td>
<td></td>
<td>Specifies the objects to add to the SDF project. When specified, ensure that you do not use the <code>-all</code> option. Two object reference formats are supported. The recommended format consists of attribute=value pairs. For example, the following is a custom object reference to an object with a script ID of custobject_sample:</td>
</tr>
<tr>
<td><code>-h</code> or <code>-help</code></td>
<td></td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>

### Example

The following command updates the manifest file of an SDF project. The command adds dependencies on the CUSTOMRECORD feature, a file, and an object.

```
sdfcli adddependencies
    -account 0000001 -email test@netsuiteide.com -p c:/project/
    -role 1014 -url system.netsuite.com -feature CUSTOMRECORD:required -file /js/test.js
    -object "scriptid=custobject_sample"
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to **Setup > Users/Roles > Manage Roles** to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic **Enabling the Show Internal IDs Preference**.

When executed, the command prompts you to update the manifest file and produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Manifest file will be updated to add the dependencies, do you want to continue?
Type YES to update the manifest file.
YES
Done.
```
authenticate

Grants authorization for SuiteCloud SDK to access NetSuite with an account-role combination. When you run this command in CLI for Java, a browser prompts so that you can log in to the role you want to use in the tool. Once the browser is open, you can issue a TBA token in NetSuite and it gets saved to use in the CLI for Java.

**Important:** In order for this command to work, at least one of the following ports needs to be accessible locally from your machine: 52300 - 52315.

**Syntax**

```
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure. When specified, the account ID value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-authid</td>
<td>Optional If you do not specify it, the default account is used. References the custom name you gave to a specific account-role combination. If you do not have an authentication ID (authid) for a specific account-role combination, you create it by passing it as an option. If you do, you cannot combine the -authid option with the following options: -email, -account, -url, or -role.</td>
<td></td>
</tr>
<tr>
<td>-savetoken</td>
<td>Optional Enter a TBA token that you issued previously in NetSuite.</td>
<td></td>
</tr>
<tr>
<td>-tokenid</td>
<td>Required if the -savetoken option is specified. Enter the token ID of a TBA token you issued previously in NetSuite.</td>
<td></td>
</tr>
<tr>
<td>-tokensecret</td>
<td>Required if the -savetoken option is specified. Enter the token secret of a TBA token you issued previously in NetSuite.</td>
<td></td>
</tr>
<tr>
<td>-url</td>
<td>Optional References the URL used to access the account. If it is not specified, the production URL is used. That is, a URL such as &quot;system.netsuite.com&quot;.</td>
<td></td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional Print the help for this command.</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

The following command authenticates an account by prompting the browser:
sdfcli authenticate -authid myDevAccount

After the browser closes and the account-role combination is authenticated, the command produces output that is similar to the following:

Authenticating...
The TBA token was successfully saved.

createproject

Creates an SDF project, either an account customization project (ACP) or a SuiteApp project.

Syntax

createproject [-overwrite]  
-type AccountType  
-parentdirectory FolderPath  
-projectname ProjectName  
[-projectid ProjectID]  
[-publisherid PublisherID]  
[-projectversion ProjectVersion]

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-type</td>
<td>Required</td>
<td>The project type. Valid values are ACCOUNTCUSTOMIZATION or SUITEAPP.</td>
</tr>
<tr>
<td>-overwrite</td>
<td>Optional</td>
<td>Indicates that an existing project with an identical project name can be overwritten.</td>
</tr>
</tbody>
</table>
| -parentdirectory | Required when type is specified as SUITEAPP | Sets the parent directory of the project created.  
For example:  
c:/Users/kwolfe/Desktop/NetSuiteCustomizations  

**Note:** You must have read, write, and execute access on any directory where you want to run CLI.

| -projectname  | Required            | A name for your project.                                                   |
|---------------|---------------------|                                                                            |
| -projectid    | Required when type is specified as SUITEAPP | A unique identifier to assign to the project.  
This value must be a word with lowercase alphanumeric characters. It must be less than 40 characters |
| -publisherid  | Required when type is specified as SUITEAPP | The publisher ID of your SuiteApp project.                                 |
CLI for Java Commands

- **projectversion**
  Required when type is specified as SUITEAPP
  The version to assign to your project, using semantic versioning. For example, specify the following project version if your SDF SuiteApp is version 1.02, minor revision #01242017:

  1.02.01242016

  **Important:** After you deploy the SDF SuiteApp to your NetSuite account, the project version number cannot be modified. The project version number of all subsequent deployments of the SDF SuiteApp must match the project version number of the initial deployment.

- **-h or -help**
  Optional
  Print the help for this command.

## Example

The following command creates an account customization project:

```
sdfcli createproject
  [-overwrite] -type ACCOUNTCUSTOMIZATION -parentdirectory d:/myprojects/customizations -projectname MySDFProject
```

The following command creates a SuiteApp project:

```
sdfcli createproject
  [-overwrite] -type SUITEAPP -parentdirectory d:/sdfprojects/mysuiteapps -projectid mysuiteapp123 -projectname MySDFProject2 -publisherid com.netsuite -projectversion 1.0.0
```

## deploy

Deploys the folder or zip file that contains the SDF project.

The SDF project folder is compressed into a ZIP file before it is deployed. Only the files and folders referenced in the deploy.xml file are included in the ZIP file that is deployed.

**Important:** For installation script, installation preference, and translation collection objects, files referenced within their xml structure are also deployed. Even if these files are not part of the deploy.xml file.

## Syntax

```
deploy [-account AccountID]
  [-authid myNetSuiteDevRole]
  [-email EmailAddress]
  [-p FolderOrZipFile]
  [-role InternalID]
  [-url NetSuiteDomainURL]
  [-applycontentprotection (T|F)]
  [-l LogPathOrFile]
  [-np]
  [-v]
  [-sw]
```
## Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-account</code></td>
<td>Required when an account ID is not specified in the SDF file.</td>
<td>References the ID of the account to log in to.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td><code>-authid</code></td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify it, the default account is used.</td>
<td>When you use it, you should not specify any of the following options: <code>-email</code>, <code>-account</code>, <code>-url</code>, or <code>-role</code>.</td>
</tr>
<tr>
<td><code>-email</code></td>
<td>Required when an e-mail address is not specified in the SDF file.</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>When specified, the email address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td><code>-p</code></td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td><code>-role</code></td>
<td>Required when a role ID is not specified in the SDF file.</td>
<td>References the internal ID of the role to use when logged in.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>This value should be set to the internal ID of a custom role that has SDF permissions assigned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When specified, the role ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td><code>-url</code></td>
<td>Required when a URL is not specified in the SDF file.</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>When specified, the URL value in the .sdf file is ignored.</td>
</tr>
<tr>
<td><code>-applycontentprotection</code></td>
<td>Required when locking.xml or hiding.xml exists in the InstallationPreferences folder of a SuiteApp project.</td>
<td>Indicates whether content protection should be applied.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the help topic Installation Preferences in a SuiteApp Project</td>
<td>For more information, see the help topic Installation Preferences in a SuiteApp Project</td>
</tr>
<tr>
<td><code>-accountsricvalues</code></td>
<td>Optional</td>
<td>Indicates how to handle the presence of account-specific values in an account customization project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the argument value is WARNING and there are account-specific values in the</td>
</tr>
</tbody>
</table>
project, a warning message is thrown but the deployment can continue.

If the argument value is ERROR and there are account-specific values in the project, and error message is thrown and deployment stops.

For more information, see the help topic Account-Specific Values.

**Important:** If the –accountspecificvalues option is not specified, the behavior is the same as passing ERROR as an argument.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-l</code></td>
<td>Optional Specifies the deployment log file location as a directory path or a file name. If you specify a directory, the log file is generated in the specified location by default. Deployment log details are appended to the end of the file.</td>
</tr>
<tr>
<td><code>-np</code></td>
<td>Optional Skips the preview step.</td>
</tr>
<tr>
<td><code>-v</code></td>
<td>Optional Starts client-side project validation. The deployment stops when any errors occur during this process.</td>
</tr>
<tr>
<td><code>-sw</code></td>
<td>Optional Removes the warning prompted before deploying to a production account.</td>
</tr>
<tr>
<td><code>-h</code> or <code>-help</code></td>
<td>Optional Print the help for this command.</td>
</tr>
</tbody>
</table>

**Example**

The following command deploys an SDF project to an account and creates a log of the deployment process:

```
sdfcli deploy
    -account 0000001 -email test@netsuiteide.com -p c:/project/ -role 1014 -url system.netsuite.com -l d:/path/test.log -np -v
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
    Deploy 'c:\project'.
2019-08-02 11:08:23 (PST) Installation started
Info -- Account [(PRODUCTION) Wolfe Electronics]
```
importbundle

Imports a customization bundle from your NetSuite account and converts it to an account customization project.

Syntax

importbundle -bundleid BundleID
[-authid myNetSuiteDevRole]
[-account AccountID]
[-email EmailAddress]
[-p FolderOrZipFile]
[-role InternalID]
[-url NetSuiteDomainURL]

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-bundleid</td>
<td>Required</td>
<td>Specifies the ID of the bundle to import.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. For more information, see SDF File Structure.</td>
<td>References the e-mail address to use to log in to the account. When specified, the e-mail address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
</tbody>
</table>
want to specify a project other than the default. For more information, see project.

-\texttt{role} \hspace{2cm} \textbf{Required when a role ID is not specified in the SDF file.}

For more information, see \texttt{SDF File Structure}.

References the internal ID of the role to use when logged in.

\texttt{This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.}

-\texttt{url} \hspace{2cm} \textbf{Required when a URL is not specified in the SDF file.}

For more information, see \texttt{SDF File Structure}.

References the URL used to access the NetSuite account.

\texttt{When specified, the URL value in the .sdf file is ignored.}

-\texttt{h or -help} \hspace{2cm} \textbf{Optional}

Print the help for this command.

\textbf{Example}

\textbf{The following command imports a bundle and creates an account customization project:}

\begin{verbatim}
  sdfcli importbundle
  -bundleid 123 -account 0000001 -email test@netsuiteide.com
  -p c:/project/ -role 1014 -url system.netsuite.com
\end{verbatim}

\textbf{Note:} In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the \textbf{Show Internal IDs} preference in your account. Go to \texttt{Setup > Users/Roles > Manage Roles} to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic \textbf{Enabling the Show Internal IDs Preference}.

When executed, the command produces output that is similar to the following:

\begin{verbatim}
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password: 
Existing files will be overwritten, do you want to continue? Type YES to continue. YES
Importing bundle with id 123 to path: "C:\project".

The following bundle component(s) were imported. 
Objects\customlist_sample_cust_list_1.xml
\end{verbatim}

\textbf{importconfiguration}

Imports the feature configuration from your NetSuite account to the account customization project so that you can enable or disable features in the project.
Syntax

```
importconfiguration -configurationid ConfigID
       [-authid myNetSuiteDevRole]
       [-account AccountID]
       [-email EmailAddress]
       [-p FolderOrZipFile]
       [-role InternalID]
       [-url NetSuiteDomainURL]
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-configurationid</td>
<td>Required</td>
<td>Specifies the configuration to import. This value should be set to FEATURES:ALL_FEATURES. You can only import the configuration for all features.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the email address to use to log in to the account. When specified, the email address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file.</td>
<td>References the URL used to access the NetSuite account. When specified, the URL value in the .sdf file is ignored.</td>
</tr>
</tbody>
</table>
CLI for Java Commands

For more information, see .SDF File Structure.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h or -help</td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>

### Example

The following command imports the feature configuration from your NetSuite account to an account customization project:

```bash
sdfcli importconfiguration
-configurationid FEATURES:ALL_FEATURES -account 0000001 -email test@netsuiteide.com
-p c:/project/ -role 1014 -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
The following object(s) were imported successfully:
FEATURES:ALL_FEATURES
```

importfiles

Imports files from your NetSuite account to the account customization project.

### Syntax

```bash
importfiles -paths FileCabinetPath1 [FilePathCabinet2] [...] 
[-authid myNetSuiteDevRole] 
[-account AccountID] 
[-email EmailAddress] 
[-p FolderOrZipFile] 
[-role InternalID] 
[-url NetSuiteDomainURL] 
[-excludeproperties]
```

### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-paths</td>
<td>Required</td>
<td>Specifies the file cabinet path or paths that contain the files to import.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-authid</td>
<td>Optional. If you do not specify it, the default account is used. References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
<td></td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. References the email address to use to log in to the account. When specified, the email address value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. References the folder or zip file that contains the SDF project.</td>
<td></td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file. References the URL used to access the NetSuite account. When specified, the URL value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-excludeproperties</td>
<td>Optional. Exclude the file and folder properties specified in your NetSuite account. If you specify this option, the .attributes folder is not created.</td>
<td></td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional. Print the help for this command.</td>
<td></td>
</tr>
</tbody>
</table>

### Example

The following command imports a file from a NetSuite account into a specified SDF project:

```
$sdfcli importfiles
   -paths "~/SuiteScripts/test.js" -account 0000001
   -email test@netsuiteide.com -p c:/project/
   -role 1014 -url system.netsuite.com
```
**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command prompts you to overwrite any existing files in the SDF project and produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Existing files will be overwritten, do you want to continue? Type YES to continue.
YES
Done
```

**importobjects**

Import custom objects from an account to your SuiteCloud project. By default in account customization projects (ACP), if SuiteScript files are referenced in custom objects you import, these files are also imported.

**Syntax**

```
importobjects <-destinationfolder DestinationPath>
    <-scriptid ScriptID1 [ScriptID2] [...]>
    <-type CustomObjectType>
    [-account AccountID]
    [-authid myNetSuiteDevRole]
    [-email EmailAddress]
    [-p FolderOrZipFile]
    [-role InternalID]
    [-url NetSuiteDomainURL]
    [-appid QualifiedAppID]
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-destinationfolder</td>
<td>Required</td>
<td>Indicates where custom objects are saved. Specify a folder that is relative to the SDF project. This value must start with &quot;/Objects&quot;.</td>
</tr>
<tr>
<td>-scriptid</td>
<td>Required</td>
<td>Imports custom objects that are associated with the specified scripts IDs.</td>
</tr>
<tr>
<td>-type</td>
<td>Required</td>
<td>Imports custom objects that are associated with the specified custom script types. Specify the ALL value to import custom objects of all custom script types.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. References the ID of the account to log in to.</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-authid</td>
<td>Optional. If you do not specify it, the default account is used. If you use this option, you should not specify any of the following options: -email, -account, -url, or -role.</td>
<td></td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. References the email address to use to log in to the account. When specified, the email address value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. References the folder or zip file that contains the SDF project.</td>
<td></td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file. References the URL used to access the NetSuite account. When specified, the URL value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-appid</td>
<td>Optional. References the application ID of the SDF SuiteApp. The application ID is a fully qualified name with the notation: &lt;publisherid&gt;.&lt;projectid&gt;. For example, com.netsuite.mysuiteapp or org.mycompany.helloworld are valid app IDs. When application ID is specified, only custom objects that are associated with the app ID are imported. Otherwise, only custom objects that are not associated with an app ID are imported. For information about the SuiteApp application ID, see the help topic SuiteApp Application ID.</td>
<td></td>
</tr>
<tr>
<td>-excludelfiles</td>
<td>Optional. Indicates that the SuiteScript files referenced in custom objects are not imported. It can only be used in account customization projects (ACP). If this option is not specified, the SuiteScript files referenced in the imported custom objects are imported too.</td>
<td></td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional. Print the help for this command.</td>
<td></td>
</tr>
</tbody>
</table>
Example

The following command imports two customrecordtype objects, customlist2 and customrole1009, into a project.

```
sdfcli importobjects
   -destinationfolder "/Objects/subfolder"
   -scriptid customlist2 customrole1009
   -type ALL -account 00000001
   -email test@netsuiteide.com -p c:/project/
   -role 1014 -url system.netsuite.com
   -appid org.mycompany.helloworldapp
```

Note: In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command prompts you to overwrite any existing objects in the SDF project and produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Existing objects will be overwritten, do you want to continue? Type YES to continue.
YES
The following object(s) were imported successfully:
customlist:customlist2
role:customrole1009
```

issuetoken

Issues a TBA token to authenticate against your account.

For more information, see Token-based Authentication in CLI for Java.

Syntax

```
issuetoken <-account AccountID>
   [-authid myNetSuiteDevRole]
   <-email EmailAddress>
   <-role InternalID>
   <-url NetSuiteDomainURL>
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
</tbody>
</table>
CLI for Java Commands

<table>
<thead>
<tr>
<th>Option</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination. If you do not have an authentication ID (authid) for a specific account-role combination, you create it by passing it as an option. If you do, you cannot combine the -authid option with the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account. Note: This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in. Note: This command does not read from SDF files. You must explicitly specify this option at the command line. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account. Note: This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
</tbody>
</table>

-h or -help  Optional
Print the help for this command.

Example
The following command issues a token to a NetSuite account:

```
sdfcli issuetoken
   -account 0000001 -email test@netsuiteide.com
   -role 1014 -url system.netsuite.com
```

Note: In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:
listbundles

Lists the customization bundles that were created in your NetSuite account.

Syntax

```
listbundles <-account AccountID>
 [-authid myNetSuiteDevRole]
 <-email EmailAddress>
 <-role InternalID>
 <-url NetSuiteDomainURL>
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify it, the default account is used.</td>
<td>When you use it, you should not specify any of the following options: <code>-email</code>, <code>-account</code>, <code>-url</code>, or <code>-role</code>.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
<td></td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
<td>This value should be set to the internal ID of a custom role that has SDF permissions assigned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
</tbody>
</table>
Note: This command does not read from SDF files. You must explicitly specify this option at the command line.

-h or -help   Optional     Print the help for this command.

Example

The following command lists all the bundles in a NetSuite account:

```shell
sfdcli listbundles
   -account 0000001 -email test@netsuiteide.com
   -role 1014 -url system.netsuite.com
```

Note: In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```text
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
123456:My Test Bundle
123456:Managed Bundle to Production
```

listconfiguration

Lists the configuration types that you can import from your NetSuite account.

Syntax

```shell
listconfiguration <-account AccountID>
   [-authid myNetSuiteDevRole]
   <-email EmailAddress>
   <-role InternalID>
   <-url NetSuiteDomainURL>
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
</tbody>
</table>
### CLI for Java Commands

#### Note:
This command does not read from SDF files. You must explicitly specify this option at the command line.

<table>
<thead>
<tr>
<th>Option</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination. When you use it, you should not specify any of the following options: <code>-email</code>, <code>-account</code>, <code>-url</code>, or <code>-role</code>.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>

### Example

The following command lists the configuration types that you can import from your account:

```bash
sdfcli listconfiguration
  -account 00000001 -email test@netsuiteide.com
  -role 1014 -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
FEATURES:ALL_FEATURES:Enable Features
```
listfiles

 Lists the files in the File Cabinet of your NetSuite account.

Syntax

```
listfiles <-account AccountID>
[-authid myNetSuiteDevRole]
<-email EmailAddress>
<-folder FileCabinetPath1 [FileCabinetPath2] [...]>
<-role InternalID>
<-url NetSuiteDomainURL>
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-folder</td>
<td>Required</td>
<td>Specifies the file cabinet folder that contains the files to list. In addition to the files in the folder, the files in any subfolders are listed.</td>
</tr>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
</tbody>
</table>
**Note:** This command does not read from SDF files. You must explicitly specify this option at the command line.

- `-h` or `-help` Optional Print the help for this command.

### Example

The following command lists the files in the File Cabinet of your NetSuite account:

```
    sdfcli listfiles
    -folder "'/SuiteScripts'" -account 0000001
    -email test@netsuiteide.com
    -role 1014 -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
/SuiteScripts/file1.js
/SuiteScripts/file2.js
/SuiteScripts/file3.js
/SuiteScripts/file4.js
/SuiteScripts/RunTimeFiles/document.pdf
/SuiteScripts/RunTimeFiles/data.csv
/SuiteScripts/RunTimeFiles/log.txt
/SuiteScripts/RunTimeFiles/sample.txt
/SuiteScripts/RunTimeFiles/test.xml
```

**listmissingdependencies**

Lists the missing dependencies in the SDF project.

### Syntax

```
listmissingdependencies <-account AccountID>
    [-authid myNetSuiteDevRole]
    <-email EmailAddress>
    [-l DependencyListFileName]
    [-p FolderOrZipFile]
    <-role InternalID>
```
### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td>-l</td>
<td>Optional</td>
<td>Specifies the missing dependency log file location as a directory path or a file name. If you specify a directory, the log file is generated in the specified location by default. Missing dependency log details are appended to the end of the file.</td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>
Example

The following command lists the missing dependencies in an SDF project:

```
sdfcli listmissingdependencies
   -p c:/project/ -l test.log
```

The command specifies to output a list of missing dependencies for the project located in the directory `c:/project`, and write the list to the file `test.log`.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Unresolved dependencies:
Feature - CUSTOMRECORDS:required
Object - customsearch51
Writing to log file C:\sdfprojects\myproject\test.log.
```

The command output states that the feature CUSTOMRECORDS is required, therefore that feature must be enabled in the manifest. The object customsearch51 is a dependency that is missing from the project. To resolve the dependency, the customsearch51 object must be added to the project or referenced in the manifest. The command writes the list to `test.log`, located in the SDF CLI installation directory.

listobjects

Lists the custom objects in your NetSuite account.

Syntax

```
listobjects [-account AccountID]
   [-authid myNetSuiteDevRole]
   [-email EmailAddress]
   [-p FolderOrZipFile]
   [-role InternalID]
   [-url NetSuiteDomainURL]
   [-appid QualifiedAppID]
   [-scriptid ScriptID]
   [-type CustomObjectType1 [CustomObjectType2] [...]]
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to. The account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination. When you use it, you should not specify any of the following options: <code>-email</code>, <code>-account</code>, <code>-url</code>, or <code>-role</code>.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
</tbody>
</table>
The email address value in the .sdf file is ignored.

- **p**
  Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see `project`.

  References the folder or zip file that contains the SDF project.

- **role**
  Required

  References the internal ID of the role to use when logged in.

  This value should be set to the internal ID of a custom role that has SDF permissions assigned.

  An account administrator should assign you with an SDF Developer role before using CLI for Java. For more information, see the help topic Assigning the Developer Role (Admin Only).

  The role ID value in the .sdf file is ignored.

- **url**
  Required

  References the URL used to access the NetSuite account.

  The URL value in the .sdf file is ignored.

- **appid**
  Optional

  References the application ID of the SDF SuiteApp.

  The application ID is a fully qualified name with the notation: `<publisherid>.<projectid>`. For example, com.netsuite.mysuiteapp or org.mycompany.helloworld are valid app IDs.

  When application ID is specified, only custom objects that are associated with the app ID are listed. Otherwise, only custom objects that are not associated with an application ID are listed.

  For information about the SuiteApp application ID, see the help topic SuiteApp Application ID.

- **scriptid**
  Optional

  Lists the custom objects with script IDs that match or contain the string value of the specified script ID.

  If you specify this option with the `-type` option, the list of custom objects is also checked against the criteria set by the `-type` option.

  If the `-scriptid` and `-type` options are not specified, all custom objects are listed.

- **type**
  Optional

  Lists the custom objects that are associated with the specified custom NetSuite types.

  If you specify this option with the `-scriptid` option, the list of custom objects is also checked against the criteria set by the `-scriptid` option.

  If the `-scriptid` and `-type` options are not specified, all custom objects are listed.

- **h or -help**
  Optional

  Print the help for this command.

---

**Example**

The following command lists the custom objects in your NetSuite account:

```
sdfcli listobjects
```
CLI for Java Commands

-manageauth

Manages authentication IDs (authid) for all your projects.

Syntax

manageauth

[-authid myNetSuiteDevRole -info]
[-list]
[-authid myNetSuiteDevRole -rename myOtherNetSuiteDevRole]
[-authid myNetSuiteDevRole -remove]

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-info</td>
<td>Optional</td>
<td>Prints the following information for the specified authentication ID (authID): company name, role ID, and URL.</td>
</tr>
<tr>
<td>-list</td>
<td>Optional</td>
<td>Prints a list of all the configured authentication IDs (authIDs).</td>
</tr>
<tr>
<td>-remove</td>
<td>Optional</td>
<td>Remove an authentication ID (authID). Usage: sdfcli manageauth -remove (authID).</td>
</tr>
<tr>
<td>-rename</td>
<td>Optional</td>
<td>Rename an authentication ID (authID). You must specify it together with the renameto option.</td>
</tr>
</tbody>
</table>
Usage: `sdfcli manageauth -rename {authID} -renameto {newauthID}`.

- `rename` option is specified.

Required when the `rename` option is specified.
Enter the new ID for an authentication ID (authID). You must specify it together with the `rename` option.

Usage: `sdfcli manageauth -rename {authID} -renameto {newauthID}`.

-h or -help  Optional
Print the help for this command.

Example

When you run `sdfcli manageauth -list`, the command produces output that is similar to the following:

The following authentication IDs (authID) are available for your account:
- myProductionAccount
- myDevelopmentAccount
- testingSandbox

preview

Previews the deployment steps of a folder or zip file that contains the SDF project.
The SDF project is compressed into a zip file before previewing.

Syntax


Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you give to a specific account-role combination.</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>If you do not specify it, the default account is used.</td>
</tr>
</tbody>
</table>
When you use it, you should not specify any of the following options: `-email`, `-account`, `-url`, or `-role`.

- **-email**: Required when an e-mail address is not specified in the SDF file.
  For more information, see .SDF File Structure.
  References the email address to use to log in to the account.
  When specified, the email address value in the .sdf file is ignored.

- **-p**: Required when a default project has not been specified by the project command or when you want to specify a project other than the default.
  For more information, see project.
  References the folder or zip file that contains the SDF project.

- **-role**: Required when a role ID is not specified in the SDF file.
  For more information, see .SDF File Structure.
  References the internal ID of the role to use when logged in.
  This value should be set to the internal ID of a custom role that has SDF permissions assigned.
  An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).
  When specified, the role ID value in the .sdf file is ignored.

- **-url**: Required when a URL is not specified in the SDF file.
  For more information, see .SDF File Structure.
  References the URL used to access the NetSuite account.
  When specified, the URL value in the .sdf file is ignored.

- **-applycontentprotection**: Required when locking.xml or hiding.xml exists in the InstallationPreferences folder of a SuiteApp project.
  Indicates whether content protection should be applied.
  For more information, see the help topic Installation Preferences in a SuiteApp Project.

- **-l**: Optional
  Specifies the preview log file location as a directory path or a file name.
  If you specify a directory, the log file is generated in the specified location by default.
  Preview log details are appended to the end of the file.

- **-h** or **-help** Optional
  Print the help for this command.

---

**Example**

The following command previews deployment steps for the SDF project and stores them in a log file:

```
sdfcli preview
  -account 00000001 -email test@netsuiteide.com -p c:/project/
  -role 1014 -url system.netsuite.com -l d:/path/test.log
```
Note: In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Preview "c:\project".
Validate manifest -- Success
Validate deploy file -- Success
Validate configuration -- Success
Validate script file -- Success
Validate objects -- Success
Validate resource files -- Success
Validate account settings -- Success
Deployment preview
Upload file -- ~/FileCabinet/SuiteScripts/UserEventScript.js
```

**project**

Sets the default project folder or zip file for CLI.

When you run this command without the `-p` option, the default project folder or zip file is displayed.

If a default project has been specified, you do not need to specify a project when using CLI commands.

**Syntax**

```
project [-p FolderOrZipFile]
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-p</code></td>
<td>Required when specifying a default project.</td>
<td>Sets the default folder or zip file that contains the SDF project. When specified, you do not need to specify the <code>-p</code> option in most other CLI commands unless you want to reference a different project folder or zip file.</td>
</tr>
<tr>
<td><code>-h</code> or <code>-help</code></td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>

**Example**

The following command sets a default project based on a specified path:

```
sdfcli project
   -p c:/project/
```
When executed, the command produces output that is similar to the following:

| SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0) |
| Default project folder set to c:\project. |

**revoketoken**

Revokes the latest TBA token that was issued to your account.

For more information, see [Token-based Authentication in CLI for Java](#).

**Syntax**

```
revoketoken <-account AccountID> 
[-authid myNetSuiteDevRole] 
<-email EmailAddress> 
<-role InternalID> 
<-url NetSuiteDomainURL>
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to.</td>
</tr>
</tbody>
</table>

**Note:** This command does not read from SDF files. You must explicitly specify this option at the command line.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination.</td>
</tr>
</tbody>
</table>

If you do not specify it, the default account is used. When you use it, you should not specify any of the following options: `-email`, `-account`, `-url`, or `-role`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account.</td>
</tr>
</tbody>
</table>

**Note:** This command does not read from SDF files. You must explicitly specify this option at the command line.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in.</td>
</tr>
</tbody>
</table>

**Note:** This command does not read from SDF files. You must explicitly specify this option at the command line. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic [Assigning the Developer Role (Admin Only)](#).

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account.</td>
</tr>
</tbody>
</table>
Note: This command does not read from SDF files. You must explicitly specify this option at the command line.

-h or -help  Optional  Print the help for this command.

Example

The following command revokes a token to a NetSuite account:

```sh
sdfcli revoketoken
-account 0000001 -email test@netsuiteide.com
-role 1014 -url system.netsuite.com
```

Note: In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```sh
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Token revoked.
```

savetoken

Saves the latest TBA token created from your account to use with CLI for Java. If a token was previously saved, it is overwritten.

After successfully saving the token, SDF stores the information and you can use the role for SDF features that depend on account login such as validation, deployment. SDF stores the token in the CLICACHE file in the same directory where SDK is located.

Important: TBA tokens can be assigned to multiple accounts, users, and roles, but you can only have one TBA token per account-user-role combination. For example, assume that you created three TBA tokens for an account named WolfeElectronics with a role named SDF Developer that is assigned to a user named awolfe@wolfeelectronics.com. In this scenario, the awolfe@wolfeelectronics.com user can only use the latest-issued TBA token with CLI to log in to the WolfeElectronics account. The first two TBA tokens should be manually removed from the account using the NetSuite interface. For more information, see the help topic Viewing, Editing, Creating, and Revoking TBA Tokens.

For more information, see Token-based Authentication in CLI for Java.

Syntax

```sh
savetoken <-account AccountID>
[<authid myNetSuiteDevRole>]
```
### Option Descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required</td>
<td>References the ID of the account to log in to. <strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. If you do not have an authentication ID (authid) for a specific account-role combination, you create it by passing it as an option. If you do, you cannot combine the -authid option with the following options: -email, -account, url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required</td>
<td>References the email address to use to log in to the account. <strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-role</td>
<td>Required</td>
<td>References the internal ID of the role to use when logged in. <strong>Note:</strong> This value should be set to the internal ID of a custom role that has SDF permissions assigned. <strong>Note:</strong> Before you can log in with a 2FA role from CLI for Java, you must set up a 2FA role for SDF Development. For further information, see the help topic <a href="#">Setting Up Your Role for SDF Development</a>.</td>
</tr>
<tr>
<td>-tokenid</td>
<td>Required</td>
<td>References the Token ID used to log in to the account.</td>
</tr>
<tr>
<td>-tokensecret</td>
<td>Required</td>
<td>References the Token Secret used to log in to the account.</td>
</tr>
<tr>
<td>-url</td>
<td>Required</td>
<td>References the URL used to access the NetSuite account. <strong>Note:</strong> This command does not read from SDF files. You must explicitly specify this option at the command line.</td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>
Example

The following command saves a token to SDF:

```
    sdfcli savetoken
    -account 0000001 -email test@netsuiteide.com
    -role 1014 -tokenid ABC123
    -tokensecret 321CBA -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to **Setup > Users/Roles > Manage Roles** to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic **Enabling the Show Internal IDs Preference.**

The token key and token secret from the example are not functional. For more information about issuing valid credentials, see **Token-based Authentication in CLI for Java and Getting Started with Token-based Authentication.**

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2018.2.0)
Token has been saved.
```

**update**

Updates existing custom objects in the SDF project folder with the custom objects in your NetSuite account.

**Syntax**

```
update <-scriptid scriptID1 [scriptID2] [...]>
    [-account AccountID]
    [-authid myNetSuiteDevRole]
    [-email EmailAddress]
    [-p FolderOrZipFile]
    [-role InternalID]
    [-url NetSuiteDomainURL]
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-scriptid</td>
<td>Required</td>
<td>Lists the script IDs of custom objects to update.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td></td>
</tr>
</tbody>
</table>
### CLI for Java Commands

#### -authid
Optional
If you do not specify it, the default account is used.

References the custom name you gave to a specific account-role combination.

When you use it, you should not specify any of the following options: `-email`, `-account`, `-url`, or `-role`.

#### -email
Required when an e-mail address is not specified in the SDF file.
For more information, see [SDF File Structure](#).

References the email address to use to log in to the account.

When specified, the email address value in the .sdf file is ignored.

#### -p
Required when a default project has not been specified by the `project` command or when you want to specify a project other than the default.
For more information, see `project`.

References the folder or zip file that contains the SDF project.

#### -role
Required when a role ID is not specified in the SDF file.
For more information, see [SDF File Structure](#).

References the internal ID of the role to use when logged in.

This value should be set to the internal ID of a custom role that has SDF permissions assigned.

An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic [Assigning the Developer Role (Admin Only)](#).

When specified, the role ID value in the .sdf file is ignored.

#### -url
Required when a URL is not specified in the SDF file.
For more information, see [SDF File Structure](#).

References the URL used to access the NetSuite account.

When specified, the URL value in the .sdf file is ignored.

#### -h or -help
Optional
Print the help for this command.

### Example

The following command checks for updates to a custom object in a NetSuite account, if it exists, and updates the object in the SDF project:

```bash
sdfcli update
  -s customrecord_test1 customrecord_test2
  -account 0000001 -email test@netsuiteide.com -p c:/project/
  -role 1014 -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to **Setup > Users/Roles > Manage Roles** to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic [Enabling the Show Internal IDs Preference](#).

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
```
updatecustomrecordwithininstances

Updates the custom record object and its instances.
Only the first 100 instances are updated based on their order of creation.

Syntax

```
updatecustomrecordwithininstances [-account AccountID]
  [-authid myNetSuiteDevRole]
  [-email EmailAddress]
  [-p FolderOrZipFile]
  [-role InternalID]
  [-url NetSuiteDomainURL]
  <-scriptid ScriptID>
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-scriptid</td>
<td>Required</td>
<td>The script ID of the custom record object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The script ID must start with “customrecord”.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file.</td>
<td>References the ID of the account to log in to.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify it, the default account is used.</td>
<td>When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file.</td>
<td>References the email address to use to log in to the account.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>When specified, the email address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file.</td>
<td>References the internal ID of the role to use when logged in.</td>
</tr>
<tr>
<td></td>
<td>For more information, see .SDF File Structure.</td>
<td>This value should be set to the internal ID of a custom role that has SDF permissions assigned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An account administrator should assign you with an SDF Developer role before using CLI. For more</td>
</tr>
</tbody>
</table>
CLI for Java Commands

### Example

The following command updates the custom record object and all instances of it, if they exist:

```
 sdfcli updatecustomrecordwithinstances
   -scriptid customrecord10 -account 0000001
   -p c:/project/ -role 1014 -url system.netsuite.com -email test@netsuiteide.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014, but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to **Setup > Users/Roles > Manage Roles** to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic **Enabling the Show Internal IDs Preference**.

When executed, the command prompts you to overwrite any existing custom objects in the SDF project and produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Existing Custom Record will be overwritten, do you want to continue? Type YES to continue.
YES
Custom record "customrecord10" has been updated with custom record instances included.
```

### uploadfiles

Uploads one or several files from an SDF project to the file cabinet in your account. The folder containing the files to upload is added to the account, if it did not already exist in the file cabinet.

#### Syntax

```
uploadfile [-account AccountID]
   [-authid myNetSuiteDevRole]
   [-email EmailAddress]
   [-project FolderOrZipFile]
   <-paths FilePath1 [FilePathCabinet2] [...]>
   [-role InternalID]
   [-url NetSuiteDomainURL]
```
## Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-paths</td>
<td>Required</td>
<td>Specifies the file path or paths that contain the files to upload. To specify multiple paths, enter a space between paths and enclose the entire argument in double quotes. For example, &quot;SuiteScripts/myscripts/sample.js&quot; &quot;SuiteScripts/morescripts/sample2.js&quot;.</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
<tr>
<td>-email</td>
<td>Required when an e-mail address is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the email address to use to log in to the account. When specified, the email address value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-project</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.</td>
<td>References the folder or zip file that contains the SDF project.</td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the URL used to access the account. When specified, the URL value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional</td>
<td>Print the help for this command.</td>
</tr>
</tbody>
</table>

## Example

The following command uploads a file from an SDF project into an account:

```
sdfcli uploadfiles
```
uploadfolders

Uploads all the files within one or several folders from a SDF project to the file cabinet in your account. The folder is added to the account if it did not already exist in the file cabinet.

The files under the specified folder, including its subfolders, are uploaded recursively.

Syntax

```
uploadfolders [-account AccountID]
[-authid myNetSuiteDevRole]
[-email EmailAddress]
[-project FolderOrZipFile]
<-paths FolderPath1 [FolderPath2] […]>
[-role InternalID]
[-url NetSuiteDomainURL]
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-paths</td>
<td>Required</td>
<td>Specifies the path or paths that contain the files to upload. To specify multiple paths, enter a space between paths and enclose the entire argument in double quotes. For example, “/SuiteScripts /SuiteScripts/Test”</td>
</tr>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td>References the ID of the account to log in to. When specified, the account ID value in the .sdf file is ignored.</td>
</tr>
<tr>
<td>-authid</td>
<td>Optional</td>
<td>References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
</tr>
</tbody>
</table>

Note: The path must start with a forward slash (/).
**CLI for Java Commands**

**-email**
Required when an e-mail address is not specified in the SDF file.

For more information, see .SDF File Structure.

References the email address to use to log in to the account.
When specified, the email address value in the .sdf file is ignored.

**-project**
Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project.

References the folder or zip file that contains the SDF project.

**-role**
Required when a role ID is not specified in the SDF file.

For more information, see .SDF File Structure.

References the internal ID of the role to use when logged in.
This value should be set to the internal ID of a custom role that has SDF permissions assigned.

An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only).
When specified, the role ID value in the .sdf file is ignored.

**-url**
Required when a URL is not specified in the SDF file.

For more information, see .SDF File Structure.

References the URL used to access the account.
When specified, the URL value in the .sdf file is ignored.

**-h or -help**
Optional

Print the help for this command.

### Example

The following command uploads a folder from an SDF project into an account:

```
sdfcli uploadfolder
  -paths "/SuiteScripts" -account 0000001
  -email test@netsuiteide.com -project c:/project/
  -role 1014 -url system.netsuite.com
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the **Show Internal IDs** preference in your account. Go to **Setup > Users/Roles > Manage Roles** to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic **Enabling the Show Internal IDs Preference**.

**validate**

Validates the folder or zip file that contains the SDF project.

### Syntax

```
validate [-account AccountID]
  [-authid myNetSuiteDevRole]
```
Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Required / Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-account</td>
<td>Required when an account ID is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td></td>
</tr>
<tr>
<td>-authid</td>
<td>Optional If you do not specify it, the default account is used. References the custom name you gave to a specific account-role combination. When you use it, you should not specify any of the following options: -email, -account, -url, or -role.</td>
<td></td>
</tr>
<tr>
<td>-email</td>
<td>Required when an email address is not specified in the SDF file. For more information, see .SDF File Structure.</td>
<td></td>
</tr>
<tr>
<td>-p</td>
<td>Required when a default project has not been specified by the project command or when you want to specify a project other than the default. For more information, see project. References the folder or zip file that contains the SDF project.</td>
<td></td>
</tr>
<tr>
<td>-role</td>
<td>Required when a role ID is not specified in the SDF file. For more information, see .SDF File Structure. References the internal ID of the role to use when logged in. This value should be set to the internal ID of a custom role that has SDF permissions assigned. An account administrator should assign you with an SDF Developer role before using CLI. For more information, see the help topic Assigning the Developer Role (Admin Only). When specified, the role ID value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-url</td>
<td>Required when a URL is not specified in the SDF file. For more information, see .SDF File Structure. References the URL used to access the NetSuite account. When specified, the URL value in the .sdf file is ignored.</td>
<td></td>
</tr>
<tr>
<td>-applycontentprotection</td>
<td>Required when locking.xml or hiding.xml exists in the InstallationPreferences folder of a SuiteApp project. Indicates whether content protection should be applied.</td>
<td></td>
</tr>
</tbody>
</table>
CLI for Java Commands

<table>
<thead>
<tr>
<th>CLI for Java Commands</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>For more information, see the help topic</td>
<td>Installation Preferences in a SuiteApp Project</td>
</tr>
<tr>
<td>-accountsSpecificValues</td>
<td>Optional</td>
</tr>
<tr>
<td>Indicates how to handle the presence of account-specific values in an account customization project. The default value is ERROR. If the argument value is WARNING and there are account-specific values in the project, a warning message is thrown. If the argument value is ERROR and there are account-specific values in the project, an error message is thrown. For more information, see the help topic Account-Specific Values.</td>
<td></td>
</tr>
<tr>
<td>-l</td>
<td>Optional</td>
</tr>
<tr>
<td>Specifies the validation log file location as a directory path or a file name. If you specify a directory, the log file is generated in the specified location by default. Validation log details are appended to the end of the file.</td>
<td></td>
</tr>
<tr>
<td>-server</td>
<td>Optional</td>
</tr>
<tr>
<td>Indicates that validation should be performed on the server. A password is required for validation When not specified, client-side validation is performed.</td>
<td></td>
</tr>
<tr>
<td>-h or -help</td>
<td>Optional</td>
</tr>
<tr>
<td>Print the help for this command.</td>
<td></td>
</tr>
</tbody>
</table>

**Example #1**

The following command performs a server-side validation on a project folder:

```
sdfcli validate
   -account 0000001 -email test@netsuiteide.com -p c:/project/
   -role 1014 -url system.netsuite.com -l d:/path/test.log -server
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Validating using account (0000001).
Validate manifest -- Success
```
CLI for Java Commands

Validate deploy file -- Success
Validate configuration -- Success
Validate script file -- Success
Validate objects -- Success
Validate resource files -- Success
Validate account settings -- Failed

*** ERROR ***

Validation of account settings failed.

An error occurred during account settings validation.
Details: The manifest contains a dependency on customrecord_tutorial object, but it is not in the account.

No errors found.
Writing to log file C:\path\test.log.

Example #2

The following command performs a client-side validation on a project folder:

```
sdfcli validate
    -account 00000001 -email test@netsuiteide.com -p c:/project/
    -role 1014 -url system.netsuite.com -l d:/path/test.log
```

**Note:** In this example, the internal ID of the custom SDF Developer role is assumed to be 1014 but this ID may not be the same as the one in your NetSuite account. To determine the internal ID of your custom role, enable the Show Internal IDs preference in your account. Go to Setup > Users/Roles > Manage Roles to search for the custom role that has SDF permissions assigned to you. The internal ID is listed next to the role. For more information, see the help topic Enabling the Show Internal IDs Preference.

When executed, the command produces output that is similar to the following:

```
SuiteCloud Development Framework CLI (com.netsuite.ide.core_2020.1.0)
Enter password:
Validating using client-side SDF metadata.
Updating SDF metadata.
SDF metadata updated.
Warnings for file c:\sdfprojects\SDFTutorialFromHelp\Objects\customrecord_tutorial.xml.
    - Line No. 13 - Error Message: The "enabledle" field depends on the "EXTREMELIST" feature.
    The manifest must define the "EXTREMELIST" feature as required or Optional
    - Line No. 15 - Error Message: The "enablemailmerge" field depends on the "MAILMERGE" feature.
    The manifest must define the "MAILMERGE" feature as required or Optional
Writing to log file C:\path\test.log.
```